Cloud and IoT Based Home Automation: Closed-loop Control of Appliances

[1]Dawood Ashraf Khan
[1]Assistant Professor
[1]Department of Computer Science
[1]North Campus, Kashmir University, Baramulla, India

Abstract : Cloud computing and Internet of Things (IoT), two very different technologies, are both already part of our technology ecosystem The use of these technologies is set to be a corner-stone of the applications on Internet in future. A future where Cloud and IoT are merged together (for example, amazons AWS Greengrass) can disruptive and technology enabler for many application use-cases. In this paper, we focus our attention on one such application, which leverages the integration of Cloud and IoT, called smart-thermostat. The smart-thermostat is a closed-loop application which is enabled due integration of cloud technology and IoT technology. We start analyzing and discussing cloud based closed-loop IoT, the progress that has been made, and how these issues have been tackled in literature survey. We then describe system model and finally we describe implementation details (along with algorithms) of smart-thermostat. Finally, we conclude with progress this works made and future scope.

***IndexTerms* - IoT; Cloud; Closed-loop; Home-Automation; MQTT.**

I. Introduction

IoT has got huge amount of potential for creating and evolving new intelligent applications in almost every field [1]. The applications of Internet of Things aim at making a human life comfortable. The Internet of Things can be defined as a system constituting of Sensors and actuators embedded in physical domain, which are linked through wired and wireless networks over Internet Protocol (IP).

Therefore, we have a ubiquitous computing, wireless technologies, sensing technologies, Internet Protocol (IP) and devices are mingled together in order to devise a system where the virtual or abstract world meets the real world meet and they interact continuously with each other. The very basic building block of Internet of Things is a―smart object, which integrates perception and intelligence into the normal or day-to-day objects. These smart objects are capable of collecting information from the environment as well as being connected to each other through the network to exchange data and information. IoT has a large number of protocols, which sit on top of the TCP/IP stack. However, in this paper we only discuss one of the popular application layer protocols called Message Queue Telemetry Transport Protocol.

The Internet of things or IOT has applicability in various domains such as media, environmental monitoring, infrastructure management, manufacturing, energy management system, medical system, home automation, and transportation systems. For example, smart city: with infrastructure like Parking, Lighting, Police, Law, Public Safety, Traffic, Pollution levels, Schools, Universities, Waste management linked together. As such, a connected municipal corporation and their departments can send garbage collection trucks to the specific locations, which need garbage collection, only. The data collected from such an IoT can further be used to do predictive maintenance of all the assets (as opposed to reactive maintenance). Moreover, connected agriculture, smart buildings, critical infrastructure all can benefit from IoT. In this paper, we choose to demonstrate the capabilities of closed-loop cloud based IoT system for a home automation problem, specifically a smart thermostat.

The smart home is a classic use-case of an IoT application—the thermostat being able to turn on ahead of time depending on the location of your car. What is important here, in our opinion, is the ability for smart-objects to communicate with one another and provide the user with a customized, easy experience of operation. Nevertheless, these IoT applications of IoT require smart object to be purchased. However, almost all the objects in home currently, which we would like to automate are not smart, as such lack the capability to be controlled over the Internet. Therefore, in order for us to be able to control the temperature in an existing environment without smart-objects, we propose to develop a smart-thermostat and retrofit it in the non-smart objects, which require temperature.

The smart-thermostat can revolutionize all common applications, which require temperature control. Users have the ability to remotely toggle on/off any attached object (non-smart) or set the temperature a smart-phone application or a web interface. Traditionally, in our day-to-day life, we are used to controlling the home appliances through conventional wall mounted switchboards. However, this conventional home appliance control system is not a convenient method for elders (aged people or physically challenged people). Moreover, there are multiple types of alternative control systems that are being developed for controlling the home appliances remotely; such systems include IR remote control home appliances, RF based home automation, and so on. Nevertheless, such systems lack range which IoT based device offer.

Distribution of the paper: in section I we give state-of-the art with respect to the closed-loop cloud based IoT automation implementations. In section III&IV, we give system model and implementation of the smart-thermostat. Finally, in section V, the paper concludes with discussion on outcomes and future ideas.

II. LITERATURE REVIEW

In this section, we review the literature about IoT. In [2], authors describe various architectural elements of a general cloud-based IoT, which is a hybrid (public/private) cloud approach. The approach, therein, is user-centric and general. However, the work only discusses a high-level architecture, without discussing technologies for implementation of real products. In [3], authors describe an IoT application of ambient assisted living for elderly, which is a closed-loop system, which requires human intervention (open-loop). However, the approach is not based on a usual IoT Stack and is based on RFID communication with non-smart objects, as a result, this setup is limited in range. Therefore, approach is closed-loop, therefore may not scale to a large scale automation. In [4], authors describe an IoT application in a rural setting. However, the authors only describe a model for wireless communication channel and not an end-to-end solution. In [5], the authors describe an IoT and cloud-based application. However, the authors only describe functional aspects of the system. Moreover, the described application is an open-loop system. In [6], the authors give a comprehensive description of an IoT and its application to various environments. However, the authors only describe business and architectural aspects of the system. In contrast, we describe a closed-loop system based. We use open-source [7,9] cloud based[8] technologies to implement our system. Our system has both sensing/monitoring capabilities and control capabilities.

III. SYSTEM MODEL OF THE CLOUD BASED IOT THERMOSTAT

System Modeling:

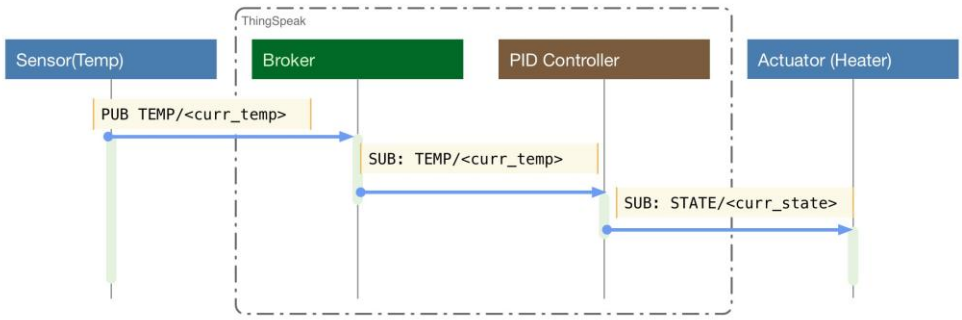

Figure 1: System model of closed-loop IoT based thermostat.

The system is consists of four major components like sensor, broker, controller and actuator, which are listed in Figure 1. The sensor is a smart-object (call it Device 1) which sense the temperature in the environment it is placed in. The Actuator is also a smart-object (call it Device 2), which actuates (turns on or off) a heater/fan, placed in the same physical environment as that of Device 1. The Device 1&2 use MQTT[10] protocol based publish-subscribe model to push or pull the state variables to or from the broker/controller. The broker is running in a cloud-based environment and receives temperature data from the Device 1(sensor). The controller is subscribed to the temperature data from Device 1 in the broker. The controller runs the PID algorithm[11] on the temperature data received from the broker and published the updated state (on or off) for the Device 2(actuator). The Device 2(actuator) is subscribed to the data (state on/off)from PID controller (also running in the cloud). Thereafter, actuator decides (based on the state update received from the PID controller) to toggles on/off the state of the connected appliance(heater/fan) accordingly. The complete closed-loop system is depicted in the Figure 2.

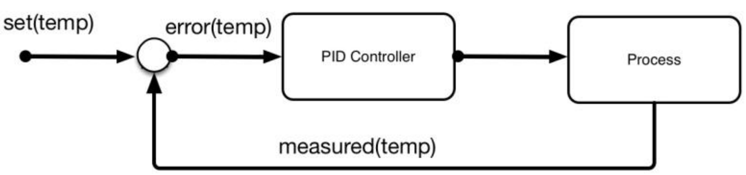

Figure 2: PID control algorithm model.

The *set(temp)* is the threshold temperature setup using a mobile application or web based application. The measured (temp) is the temperature published by Device 1(sensor) to the broker. The *error(temp)* is the difference computed by the PID controller running in the cloud of *measured(temp)* with *set(temp)*. The deviation of *error(temp)*, positive or negative, is used to decide to set the state to on/off, for the actuator.

IV. IMPLEMENTATION OF THE CLOUD BASED IOT THERMOSTAT

Implementation:

The temperature sensor read the temperature data from the system environment. The temperature sensor (we use DS18b20[12] dallas digital temperature sensor) is connected to the embedded device (expressif ESP8266[13])— described as Device 1 above. The device runs a MQTT client which publishes the data (every 1 second) to a broker running in the cloud, which is a ThingSpeak[14] channel—let us name it channel 1. The PID controller also runs in ThingSpeak, as Matlab function which is triggered every time data is received on channel 1. The Matlab function (PID algorithm) computes the state (on/off) based on the *error(temp)* and publishes the states on another ThingSpeak channel—let us name it channel 2. A heater/fan is connected to the second ESP8266 (Device 2), which run a MQTT client subscribed to channel 2. As such, Device 2 gets states updates from over the channel 2, the control algorithm running on ThingSpeak publishes to this channel. The communication between ESP8266's and ThingSpeak server is done over MQTT protocol. The ThingSpeak server holds the data, visualizes it, and also runs the control algorithm on the newly received data from the ESP8266.

ESP8266(sensor/Actuator) algorithm:

The implementation is done is C programming. The following algorithms describe algorithms on the Device 1&2:

```
/*Device 1 (sensor)*/
loop()
{
  DS18B20.requestTemperatures(); //to request DS18B20 for temperature
  tempC = DS18B20.getTempC();  //temperature in variable tempC
//has a 10 sec period passed? If Yes Send Data to ThingSpeak else continue.
  if (millis() - lastConnectionTime > postingInterval) {
    ThingSpeak.setField(1,tempC); //send tempC to channel 1
   //Write credentials for ThingSpeak channel1
    ThingSpeak.writeFields(myChannelNumber1, myWriteAPIKey1);
              lastConnectionTime = millis();
  }
      }


/*Device 2 (actuator)*/
loop()
{//have 20 secs passed? If Yes get data from Channel 2 of ThingSpeak
  if (millis() - lastReadingConnectionTime > readingInterval)
  {//Read from Thingspeak channel2 and put PID controller generated state in ONOFF.
    ONOFF = ThingSpeak.readFloatField(myChannelNumber2,1,myReadAPIKey2);
    if(ONOFF==1)
      digitalWrite(Heater, HIGH);   // turn the heater on (HIGH is the voltage level)
    else
      digitalWrite(Heater, LOW);   // turn the heater off (LOW is the voltage level)
    lastReadingConnectionTime = millis();
  }

}
```

The controller is discrete and is implemented in Matlab on ThingSpeak channel. It takes two inputs:

- *Set(temp)*: This value is set from a mobile application or web application on a ThingSpeak channel, say 40°C. This value is encoded as a fixed-point integer so that it can be compared directly with the digital temperature value from the environment model.
- State: a signal that switches the heater on or off.

The main part of the PID controller is a state machine with two top-level states:

- *Off*: The heater is off
- *On*: The heater is on.

The PID controller always remains in the *Off* state for at least 20 secs, after which it will switch to the *On* state if the temperature is less than the reference temperature. The controller may not remain in the *On* state for more than 10 secs. It will switch to the *Off* state if the temperature becomes greater than the *set(temp).*

V. CONCLUSION

This work presented a future paradigm in the IoT that is the merger of cloud+IoT technological stacks. We used this paradigm on a use-case of a smart-thermostat. We showed how IoT and Cloud can be leveraged to implement a closed-loop system.

However, we did not discuss the Quality-of-Service (QoS) issues, which can be of critical nature in closed-loop system. This point is important as the internet (IP networks) does not provide guaranteed service. This is critical for applications, which need fast response-time based temperature control. However, this work can be sufficient for applications which need slow-response time or for application which are of non-critical nature.

For future work, we would like to study cloud+IoT application paradigm for applications, which have strict QoS time constraints and low computation requirements. We would also like to study, multi-class traffic in cloud+IoT paradigm scenarios and the fairness to all traffic. As a future work, we would also like to explore the gateways, which may run a multiple Cloud+IoT stacks, in order to cater for heterogeneous setups.

REFERENCES

[1]. E. Borgia. "The Internet of Things vision: Key features, applications and open issues." Computer Communications 54 (2014): 1-31.

[2]. Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems 29, no. 7 (2013): 1645-1660.

[3]. Dohr, Angelika, Robert Modre-Osprian, Mario Drobics, Dieter Hayn, and Günter Schreier. “The Internet of Things for Ambient Assisted Living." ITNG 10 (2010): 804-809.

[4]. Rohokale, Vandana Milind, Neeli Rashmi Prasad, and Ramjee Prasad. "A cooperative Internet of Things (IoT) for rural healthcare monitoring and control." In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, pp. 1-6. IEEE, 2011.

[5]. Doukas, Charalampos, and Ilias Maglogiannis. "Bringing IoT and cloud computing towards pervasive healthcare." In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, pp. 922-926. IEEE, 2012.

[6]. Istepanian, R. S. H., Sijung Hu, N. Y. Philip, and Ala Sungoor. "The potential of Internet of m-health Things “m-IoT” for non-invasive glucose level sensing." In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5264-5266. IEEE, 2011.

[7]. Belli, Laura, Simone Cirani, Luca Davoli, Lorenzo Melegari, Màrius Mónton, and Marco Picone. "An Open-Source Cloud Architecture for Big Stream IoT Applications." Interoperability and Open-Source Solutions for the Internet of Things (2015): 73-88.

[8]. Raggett, Dave. "COMPOSE: An Open Source Cloud-Based Scalable IoT Services Platform." ERCIM News 101 (2015): 30-31.

[9]. Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.P., Riahi, M., Aberer, K., Jayaraman, P.P., Zaslavsky, A., Žarko, I.P. and Skorin-Kapov, L., 2015. Openiot: Open source internet-of-things in the cloud. Interoperability and Open-Source Solutions for the Internet of Things, pp.13-25.

[10]. “MQTT." MQTT. Accessed September 10, 2016. https://mosquitto.org.

[11]. Gaing, Zwe-Lee. "A particle swarm optimization approach for optimum design of PID controller in AVR system." IEEE transactions on energy conversion 19.2 (2004): 384-391.

[12]. DS18B20. Accessed September 10, 2016. https://www.maximintegrated.com

[13]. ESP8266. Accessed September 10, 2016. https://www.espressif.com

[14]. ThingSpeak. Accessed September 10, 2016. https://thingspeak.com